CTI概要设计

作者:祁伟

时间:2012年12月13日

**目录**

[前言： 5](#_Toc437458418)

[专业术语： 6](#_Toc437458419)

[1 引言 7](#_Toc437458420)

[1.1 编写目的 7](#_Toc437458421)

[1.2 背景 7](#_Toc437458422)

[1.3 参考资料 7](#_Toc437458423)

[2 总体设计 7](#_Toc437458424)

[2.1 需求规定 7](#_Toc437458425)

[2.2 设计原则 7](#_Toc437458426)

[2.3 设计的基本概念 8](#_Toc437458427)

[2.3.1 状态机 8](#_Toc437458428)

[2.3.1.1 外线状态机 9](#_Toc437458429)

[2.3.1.2 分机状态机 9](#_Toc437458430)

[2.3.1.3 IVR分机状态机 9](#_Toc437458431)

[2.3.1.4 软电话服务端状态机 9](#_Toc437458432)

[2.3.1.5 IVR业务流程状态机 10](#_Toc437458433)

[2.3.1.6 软电话客户端状态机 10](#_Toc437458434)

[2.3.1.7 会议设备状态机 10](#_Toc437458435)

[2.3.1.8 ACD状态机 10](#_Toc437458436)

[2.4 结构 10](#_Toc437458437)

[2.4.1 CTIServer模块： 11](#_Toc437458438)

[2.4.2 语音处理模块： 11](#_Toc437458439)

[2.4.3 ACD模块： 11](#_Toc437458440)

[2.4.4 IVRFlow模块 12](#_Toc437458441)

[2.4.5 DeviceAdapter模块 12](#_Toc437458442)

[2.4.6 数据库模块 12](#_Toc437458443)

[2.4.7 Agent模块 13](#_Toc437458444)

[2.4.8 录音模块 13](#_Toc437458445)

[2.4.8.1 并线录音模式 13](#_Toc437458446)

[2.4.8.2 会议式录音 13](#_Toc437458447)

[2.4.9 日志模块 14](#_Toc437458448)

[2.5 系统配置方式 14](#_Toc437458449)

[2.5.1 分机配置方式 14](#_Toc437458450)

[2.5.2 IVR分机配置方式 14](#_Toc437458451)

[2.5.3 Agent配置方式 15](#_Toc437458452)

[2.5.4 组配置方式 15](#_Toc437458453)

[2.5.5 坐席组配置方式 15](#_Toc437458454)

[2.5.6 ACD配置方式 16](#_Toc437458455)

[2.5.7 示例 16](#_Toc437458456)

[2.6 尚未解决的问题 17](#_Toc437458457)

[2.6.1 集群式部署 17](#_Toc437458458)

[2.6.2 数据分析 17](#_Toc437458459)

[2.6.3 智能化管理系统 17](#_Toc437458460)

[2.6.4 实时监控 17](#_Toc437458461)

[3 接口设计 18](#_Toc437458462)

[3.1 用户接口 18](#_Toc437458463)

[3.1.1 软电话坐席人员接口 18](#_Toc437458464)

[3.1.2 系统管理员接口 18](#_Toc437458465)

[3.1.2.1 分机配置管理 18](#_Toc437458466)

[3.1.2.2 IVR流程编辑 18](#_Toc437458467)

[3.1.2.3 坐席工号管理 18](#_Toc437458468)

[3.1.2.4 技能组管理 18](#_Toc437458469)

[3.1.2.5 系统监控管理 18](#_Toc437458470)

[3.1.3 配置管理接口 18](#_Toc437458471)

[3.2 外部接口 18](#_Toc437458472)

[3.2.1 软电话接口 18](#_Toc437458473)

[3.2.2 数据库查询接口 18](#_Toc437458474)

[3.3 内部接口 19](#_Toc437458475)

[3.3.1 ACD 模块接口 19](#_Toc437458476)

[3.3.2 DeviceAdapter模块接口 19](#_Toc437458477)

[3.3.3 IVR模块接口 19](#_Toc437458478)

[3.3.4 Agent模块接口 19](#_Toc437458479)

[3.3.5 数据库模块接口 19](#_Toc437458480)

[4 运行设计 19](#_Toc437458481)

[4.1 电话呼入流程 20](#_Toc437458482)

[4.2 电话外呼流程： 21](#_Toc437458483)

[5 5、数据库结构 22](#_Toc437458484)

[5.1 数据库设计原则 22](#_Toc437458485)

[5.2 数据库表列表 22](#_Toc437458486)

[5.3 数据库各表字段 22](#_Toc437458487)

[5.3.1 呼入电话记录表（InboundCallDetail） 22](#_Toc437458488)

[5.3.2 呼出电话记录表（OutboundCallDetail） 23](#_Toc437458489)

[5.3.3 呼入IVR记录表（InboundIVRDetail） 24](#_Toc437458490)

[5.3.4 转技能组记录表（TransferSkillGroupDetail） 24](#_Toc437458491)

[5.3.5 转坐席记录表（TransferAgentDetail） 25](#_Toc437458492)

[5.3.6 软电话状态明细记录表（AgentStateDetail） 25](#_Toc437458493)

[5.3.7 录音明细记录表（RecordingDetail） 26](#_Toc437458494)

[5.3.8 会议明细记录表（ConferenceDetail） 26](#_Toc437458495)

[5.4 数据库结构图 27](#_Toc437458496)

# 前言：

本文档主要目的是从可行性上设计一个通用性CTI，主要是从功能、技术理论上的设计。尽量做到框架简单，不使用复杂技术和高深设计，做到模块可添加，接口简单易用，稳定性好。可能并不适用于生产系统或发布版本。

技术来源，朱彤、李鹤峰、郭唯、岳阳。

# 专业术语：

参考《CTI需求分析》专业术语部分。

# 引言

## 编写目的

开发出适合外呼、呼入等功能的通用型CTI程序，主要应用方向适合呼叫中心、电话客服中心、IVR系统、电话销售中心、录音系统、公司办公系统。可以使用电话语音盒应用于4线以下的场合，也可以使用工控机+语音卡模式应用于50线以下场合，也可以使用交换机应用于大规模呼叫中心。

## 背景

本人毕业后即从事于呼叫中心软件开发行业，到现在有4年时间，现在将这几年的工作经验做个总结，也算是对自己这4年的一个总结。

## 参考资料

1、FreeSwitch 1.0.6 cookbook

2、数据库系统原理 经济科学出版社 主编/丁保康

3、实用软件工程 第二版 清华大学出版社 郑人杰 殷人昆 陶永磊著

4、《CTI需求分析》

5、基于有限状态机理论和工作流理论的IVR系统的设计与实现 屈金泉

# 总体设计

## 需求规定

此文档是以《CTI需求分析》为基础，主要实现上述文档中的功能需求和其他的需求。《CTI需求分析》中要求的功能主要可分为语音通话、ACD、IVR、坐席管理、数据记录、录音、性能需求、兼容性需求等。

## 设计原则

1、系统要有可靠稳定的性能，事件的处理及时、正确，数据的记录准确。

2、系统的操作接口应简单、易用、友好。

3、业务的实现要以脚本来实现，不依赖程序的代码部分，程序就好比是一个脚本解析器，处理好资源调度、线程同步、临界资源等问题，功能都在脚本中实现，修改业务只需要修改脚本和配置。

4、配置文件要以xml格式来实现，要做到分层和继承，每个程序的配置文件首先加载“程序名.XML”文件，各个实体对象的配置以面向对象方式实现，子对象的配置可以继承父对象配置，如果子对象有自己的配置将覆盖父对象的配置。

5、系统应对运行中出现的错误有充分的考虑，出现错误后要有良好的错误提示，不影响使用。做到多个通话之间无关性，一个通话的异常不会影响其他的通话。

6、考录系统的扩展性，当系统需要扩展时，只需要叠加需要的模块，其他模块可保持不变。

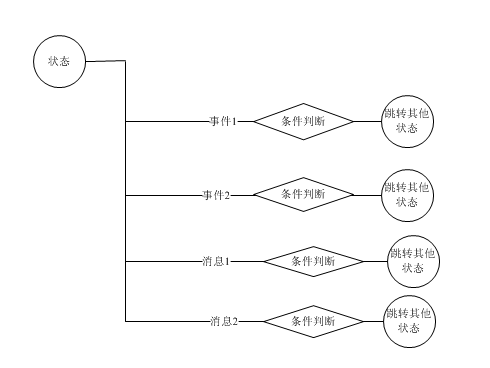
## 设计的基本概念

在CTI环境中所包含的设备、业务流程都是有状态的，业务人员的操作、设备的控制、状态变化都是在各个状态之间变化。业务流程、硬件设备在某一时刻都是处于一个状态，业务进行时一直在多个状态之间变化，此部分比较适合用状态机实现。业务逻辑的变换和修改也比较适合用状态机来描述，所以设计的基本理念是各种类型状态机之间的协同工作，包括硬件设备状态机、业务状态机两类。

将系统中的所有资源以分机描述，分机、组、技能组、坐席、IVR都虚拟成分机，所有的资源支持分机的一些基本功能，例如：转入、转出、忙、空闲、超时等。将一个电话转入IVR 可以直接将此分机转入IVR分机号，系统会将这两个资源进行连接。

### 状态机

系统中的设计以状态机为核心，各个模块和功能围绕状态机的状态改变或操作来实现。用状态机来描述设备的状态或业务逻辑，每个状态机的状态是有限的，但是每个状态机都是闭环，能从不同的状态之间转移，所以能实现无限个状态。状态机中包括状态、事件、动作、次态（下一状态）。状态结构图：



由于状态的树形结构，结合xml语言的特性，状态机使用xml语言来描述，以下是一个状态的示例。

<state id="idle">

<onentry>

<log level = "info" expr="enter idle state."/>

<send target="extension" type="cmd" idlocation="extension.number;" from="extension.number;" event="reloadConfig"></send>

</onentry>

<event event="E\_CHG\_ChState" cond="extension.eventData=='S\_CALL\_RINGING';" >

<transition target="ringing"/>

</event>

<event event="E\_CHG\_ChState" cond="extension.eventData=='S\_CALL\_STANDBY';">

<transition target="idle"/>

</event>

<onexit>

<log level = "info" expr="enter idle state."/>

</onexit>

</state>

state：状态元素

onentry:进入状态执行元素

onexit:离开状态执行元素

event:事件元素

log：日志元素

transition：跳转元素

以上6个元素可以实现状态机的主要功能，如需要其他的功能可以扩展其他的元素。

根据系统中的实体对象，计划分为八种类型状态机。

#### 外线状态机

每个外线配置一个状态机，此种类型状态机的数量是已知的，无论当前外线收到任何事件，处于何种状态，都会处于此状态机中，此状态机在程序启动后所有的生命周期都存在。

#### 分机状态机

每个配置的分机启用一个状态机，此状态机的数量是已知的，无论当前分机处于何种状态，都会处于此状态机中，此状态机在所有的生命周期中都存在。

#### IVR分机状态机

相当于只为IVR功能服务的外线分机，如果外线分机上可以满足IVR的媒体操作，此状态机可以和外线分机合并；每个配置的IVR分机都启用一个状态机，此状态机的数量是已知的，无论当前分机处于何种状态，都会处于此状态机中，此状态机在所有的生命周期中都存在。

#### 软电话服务端状态机

此状态机是由软件实现的状态机，并没有物理化的分机，是描述软电话在服务器端的状态，分机配置后此状态机生成，在所有的生命周期中都存在。

#### IVR业务流程状态机

此状态机是一个逻辑上的设备，并没有直接相对应的硬件，是IVR的业务流程，即IVR流程脚本，根据配置生成相应的状态机，此状态机在所有的生命周期中都存在。

#### 软电话客户端状态机

软电话客户端的状态，显示软电话的业务状态，控制软电话的界面显示等，此状态机在启动软电话程序后生成，关闭软电话后销毁。

#### 会议设备状态机

每个会议设备配置一个状态机，此种类型状态机的数量是已知的，无论当前设备收到任何事件，处于何种状态，都会处于此状态机中，此状态机在程序启动后所有的生命周期都存在。

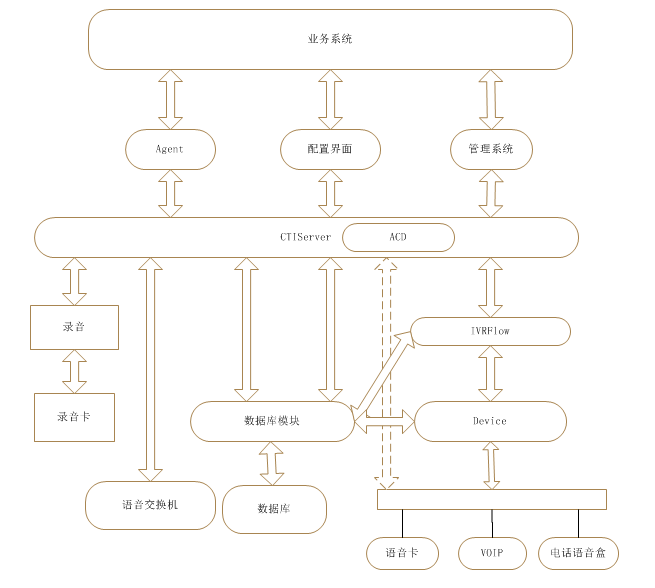
#### ACD状态机

ACD模块使用状态机来实现，ACD的业务使用状态机脚本编写，当有一个话路产生时，会新建一个ACD状态机，以后的ACD业务会在此状态机中完成，当话路销毁时通知此状态机，然后此状态机销毁。

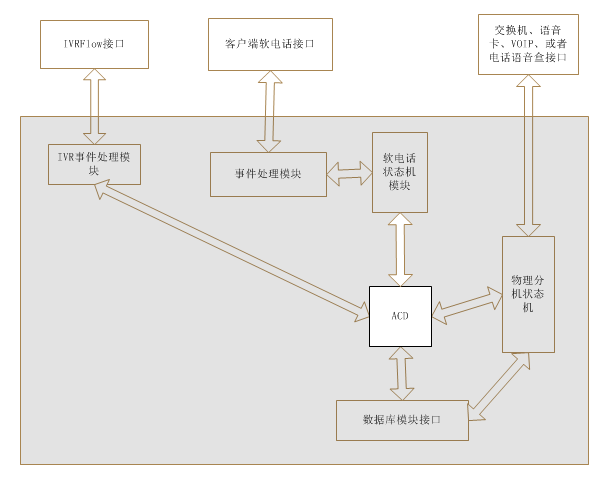
## 结构

系统中按功能的不同划分为多个模块，各个模块围绕状态机来实现，各个模块之间的通信通过传递xml消息来实现，基于xml的可扩展性，对各个模块之间的功能扩展会比较简便。各个模块将收到的消息放入消息队列中，按照顺序进行处理，处理模块根据消息来改变状态机的状态，执行状态机的动作对硬件设备进行控制。

以状态机概念为基础，结合本人这几年对CTI的理解，将系统划分为以下几个主要模块：



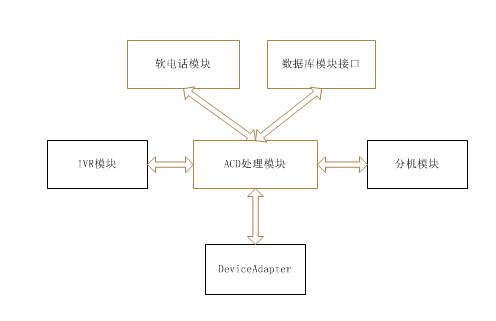
### CTIServer模块：

负责软电话的登录、管理软电话状态迁移、物理电话状态和软电话之间的逻辑同步、和软电话的交互、 和媒体模块之间消息传递（语音交换机、语音卡、VOIP服务器、语音盒等）、资源管理（分机资源、软电话资源） 。  


### 语音处理模块：

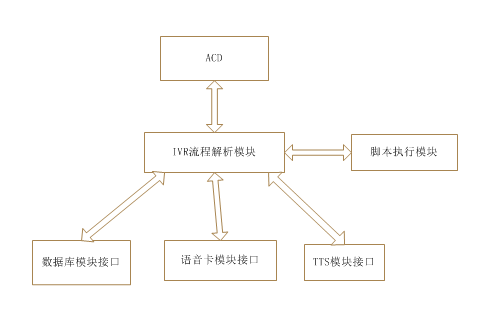
包括语音交换机、语音卡、VOIP系统、话机等，此系统中不包括此部分，实现过程中系统会调用语音处理模块的接口来控制语音和其他媒体。

### ACD模块：

此模块功能是系统中的调度核心。支持话务交换、话务路由、队列功能。此模块中所有的功能都可配置，可无限制扩充配置。  


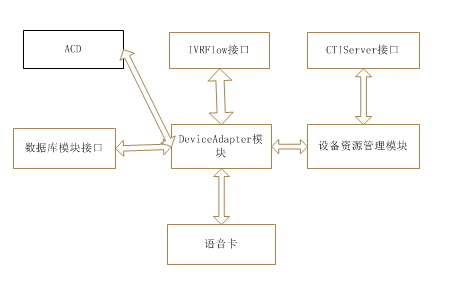
### IVRFlow模块

此模块主要解析IVR流程，根据通道事件决定流程走向，IVR功能需要调用语音卡接口、TTS接口、脚本执行模块、数据库模块，通道放音等操作。



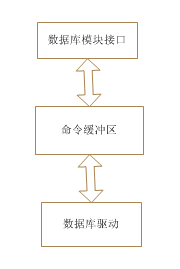
### DeviceAdapter模块

负责对媒体设备的控制，接收来自ACD、IVRFlow模块的对语音卡的命令，完成放音、收号、呼叫、来电、会议等功能，还可以保留数据库接口，完成对数据的记录。如果系统架构中没有使用交换机，此模块还和CTIServer模块之间有链接，负责语音交换机的分机配置、话路交换、会议、组等功能。



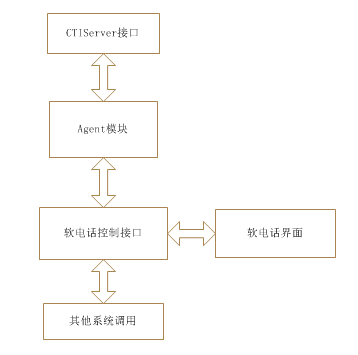
### 数据库模块

负责接收来自其他模块的数据记录命令，将数据记录进数据库中。



### Agent模块

软电话客户端模块，负责CTIServer、操作人员之间的连接，和对应的物理电话状态同步，返回所有的电话状态事件，以软件形势控制物理电话。提供和其他系统耦合的接口和坐席人员操作接口。



### 录音模块

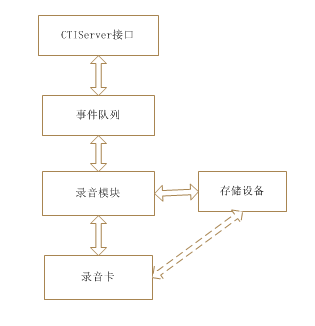
受CTI控制，启动录音设备进行录音，将录音文件存储在指定的设备中，并将录音结果返回给CTI。录音方式可以使用并线录音、会议录音两种方式。

#### 并线录音模式

采用高阻录音设备和分机进行并联，每个话机链接一个录音设备，当需要录音时启动录音设备。

#### 会议式录音

将分机和录音设备都接入交换机或语音卡中，当需要录音时创建一个会议，将分机、录音设备都加入，启动录音设备进行录音，采用此方式录音，并不需要每个分机配置一个录音设备，只需要在录音时选用其中一个录音设备。



### 日志模块

日志模块并不是一个单独运行的模块，在代码的各个部分都和日志模块都有耦合，在代码中需要记录日志的地方加入日志输出代码，日志模块可以收集此部分输出到一定地方，可以输出到屏幕、输出到文件、输出到端口。

## 系统配置方式

每个进程启动时会加载一个xml格式的配置文件，配置文件为“程序名.xml”，配置文件定义可以分为ACD、分机组（Groups）、分机（Extensions）、IVRExtensions（IVR分机）、坐席组（AgentGroups）、坐席（Agents）。

### 分机配置方式

<Extensions>

<Extension ExtensionNumber="1000" ChannelID="0" Type="Trunk" StateMachine="./config/Analog\_Trunk.xml"></Extension>

</Extensions>

Extension: 此元素是分机元素，表明是一个分机的配置信息。

ExtensionNumber:此分机的分机号码

ChannelID:此分机设备的channel ID。

Type:此分机的设备类型，类型由硬件决定主要有中继（Trunk）、分机（Extension）、传真（FAX）等类型。

StateMachine：状态机脚本文件。

### IVR分机配置方式

<IVRExtensions>

<Extension ExtensionNumber="5000" Type="IVR" StateMachine="./config/IVR.xml"></Extension>

</IVRExtensions>

Extension: 此元素是分机元素，表明是一个分机的配置信息。

ExtensionNumber:此分机的分机号码

Type:此分机的设备类型IVR类型。

StateMachine：状态机脚本文件。

### Agent配置方式

<Agents>

<Agent Id=”2000” Name=”” Password=”” StateMachine=”../config/Agent.xml” ></Agent>

</Agents>

Agent：表明此元素是一个坐席配置。

Id：坐席 id，坐席的唯一标识。

Name：坐席名称

Password:坐席密码

StateMachine：状态机脚本文件。

### 组配置方式

<Groups>

<Group ExtensionNumber ="5000" name ="IVR" >

<Extension>5001</Extension>

<Extension>5002</Extension>

</Group>

</Groups>

Group:表明此元素是 组配置信息。

ExtensionNumber：组号

Name:组名

子元素：组下的子元素是组所包含的分机号码。如<Extension>5001</Extension>表示此组中包括5001分机。

### 坐席组配置方式

<AgentGroups>

<AgentGroup ExtensionNumber=”6000” name=””>

<Agent>2000<Agent>

</AgentGroup>

</AgentGroups>

AgentGroup：表明此元素是坐席组的配置信息。

ExtensionNumber：技能组号。

Name:技能组名称

子元素：组下的子元素是技能组所包含的坐席号码，如<Agent>2000<Agent> 表示此组中包括坐席2000。

### ACD配置方式

<ACDs>

<ACD ExtensionNumber="" StateMachine="./config/ACD.xml"></ACD>

<ACDs>

ExtensionNumber:未使用字段

StateMachine：状态机脚本文件。

### 示例

以下是一个配置文件样例：

<?xml version="1.0" encoding="UTF-8"?>

<chilli>

<config>

<Extensions>

<Extension ExtensionNumber="1000" ChannelID="0" Type="Trunk" StateMachine="./config/Analog\_Trunk.xml"></Extension>

<Extension ExtensionNumber="1001" ChannelID="1" Type="Extension" StateMachine="./config/Anolog\_User.xml"></Extension>

</Extensions>

<IVRExtensions>

<Extension ExtensionNumber="5001" Type="IVR" StateMachine="./config/IVR.xml"></Extension>

<Extension ExtensionNumber="5002" Type="IVR" StateMachine="./config/IVR.xml"></Extension>

</IVRExtensions>

<Groups>

<Group ExtensionNumber ="5000" name ="IVR" >

<Extension>5001</Extension>

<Extension>5002</Extension>

<Extension>5003</Extension>

<Extension>5004</Extension>

<Extension>5005</Extension>

</Groups>

<Agents>

<Agent id=”2000” name=”” password=”” StateMachine="./config/Agent.xml ></Agent>

</Agents>

<AgentGroups>

<AgentGroup ExtensionNumber=”6000” name=”” >

<Agent>2000<Agent>

</ AgentGroup>

</AgentGroups>

<ACDs>

<ACD ExtensionNumber="" StateMachine="./config/ACD.xml"></ACD>

</ACDs>

</config>

</chilli>

## 尚未解决的问题

### 集群式部署

可以同时部署多个功能相同的服务器，服务器之间负载均衡，当其中一个出现问题时，业务自动转移到其他服务器上。

### 数据分析

将CTI 中记录的基本数据进行统计，反映出系统的硬件使用率和管理方面的概况。

### 智能化管理系统

可以帮助系统管理人员智能的进行管理和提示。可以对系统中的坐席排班，技能组划分，未接电话提示等进行辅助性提示。

### 实时监控

可以提供一个模块收集各个状态机的状态，并提供主动通知和查询功能，实现《CTI需求分析》3.11 实时监控。

# 接口设计

## 用户接口

### 软电话坐席人员接口

参考《CTI需求分析》3.5 软电话

### 系统管理员接口

#### 分机配置管理

系统中的配置文件以xml实现，修改配置文件时，可以直接修改xml文件，或者通过其他方式修改xml文件，修改完成后，重新加载配置文件生效。

#### IVR流程编辑

IVR流程主要以xml文件构成，签入脚本文件，此脚本编辑人工直接修改比较麻烦，不直观，需要提供可视化编辑器，实现方式为用Web版本，打开编辑器后可以加载系统中存在的IVR流程，编辑界面模仿VS界面，左侧是结构区，以树形结构显示出IVR的功能节点，右侧是功能编辑区，选中一个功能后，可以在右侧编辑他的属性。在结构区一个功能节点上点击右键，可以添加其他的功能节点。编辑完成后可以检查、测试所编辑的流程。测试完成后应用新流程。

#### 坐席工号管理

系统中的配置文件以xml实现，修改配置文件时，可以直接修改xml文件，或者通过其他方式修改xml文件，修改完成后，重新加载配置文件生效。

#### 技能组管理

和坐席工号管理的修改相同。

#### 系统监控管理

可以配置一些错误发生后执行的动作，可以配置发送邮件、拨打维护人员电话。

### 配置管理接口

系统的配置功能都是以xml实现，系统启动后加载xml配置，系统运行过程中需要修改配置也可以直接修改xml文件，然后执行重新加载配置文件的命令，使配置生效，但这些操作需要在服务器上操作，系统需要提供一个端口，可以让客户端登陆，登陆后可以执行配置功能命令。

## 外部接口

### 软电话接口

提供3．1．1 软电话坐席人员接口的所有功能，以API方式提供，方便和其他程序的耦合。

### 数据库查询接口

系统中的数据记录完成后需要供业务人员查询、分析。因此需要定义出数据库的查询接口，对数据库中的表不允许程序直接访问，都以存储过程方式间接访问，对数据库中的访问用户设置多种权限，每个用户下可以允许相应的存储过程。

## 内部接口

内部接口是系统中各个模块之间互相通信使用的接口，其他开发人员也可以根据此接口进行扩展功能的开发。消息体使用xml格式实现，消息类型分为命令、事件、消息三种类型。

如：<CMD Agent=”1000”>Answer</CMD>

<EVENT extenstion=”1000”>offline</EVENT>

<message Agent=”1000”>警告:对方电话忙</message>

各个内部模块保持一个消息队列，向此模块交互时需要将消息发送到此队列，此模块的执行线程会获取此消息执行。

### ACD 模块接口

### DeviceAdapter模块接口

### IVR模块接口

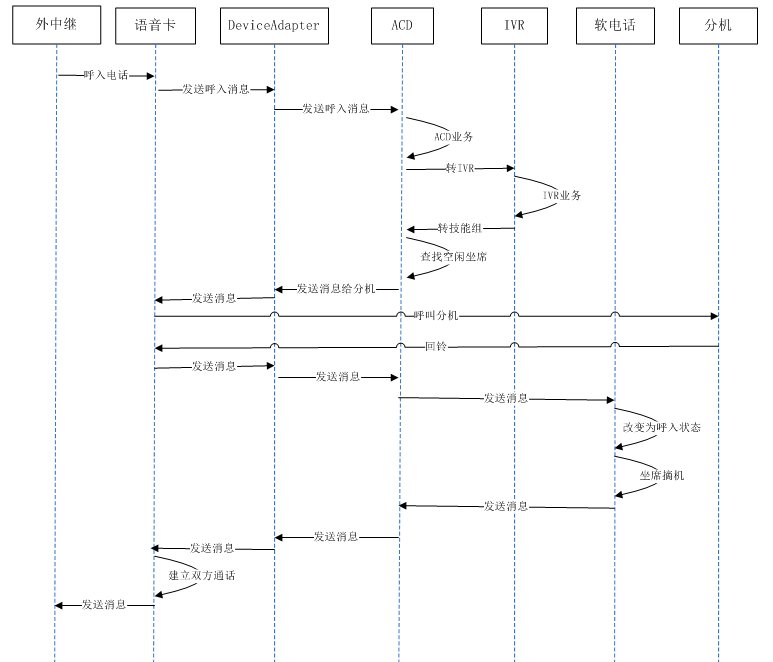
### Agent模块接口

### 数据库模块接口

# 运行设计

运行设计是针对程序在运行过程中的数据流向和对数据的处理方法，是程序的实时动态，包括统中的业务和各个模块、状态机之间的配合。

## 电话呼入流程



步骤：

1、中继上有电话呼入。

2、语音卡上收到呼叫。

3、语音卡上的消息被DeviceAdapter获取，并传送给ACD。

4、ACD模块根据主被叫号码等信息，进入ACD业务。

5、如果是转IVR的电话，转IVR。

6、电话进去IVR流程，如果客户操作了转人工技能组，发送消息给ACD。

7、ACD根据收到的消息，查找可以转接的坐席，并对坐席的分机进行振铃，将消息发送到DeviceAdapter。

9、DeviceAdapter控制语音卡，对分机振铃。

10、分机回铃。

11、语音卡将分机回铃消息发送给DeviceAdapter。

12、DeviceAdapter将分机回铃消息发送给ACD。

13、ACD收到分机回铃消息后，改变软电话状态机，并将消息发送给软电话客户端。

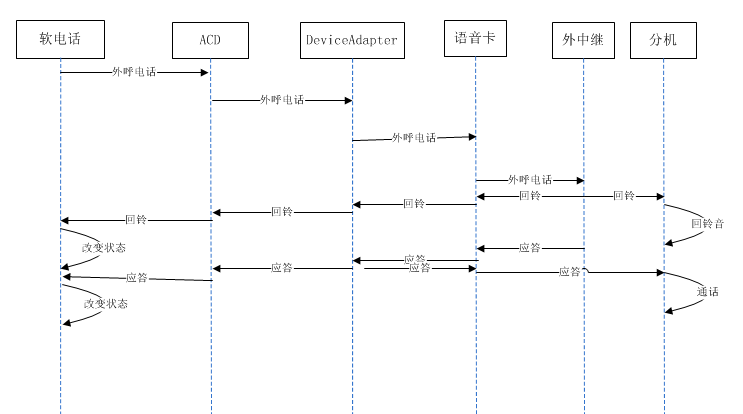
14、坐席点击“应答”按钮摘机，软电话将应答消息发送给ACD。

15、ACD收到消息后，改变软电话状态机，和分机状态机，将应答消息发送给DeviceAdapter。

16、DeviceAdapter将消息解析后，控制语音卡摘机并和外线建立双方通话。

17、语音卡执行命令，将分机和外线建立双向链接。

## 电话外呼流程：



步骤：

1、坐席操作软电话，输入电话号码后进行外呼，软电话将消息发送到ACD。

2、经ACD路由后选择外呼参数，将外呼消息发送给DeviceAdapter。

3、DeviceAdapter收到消息后，控制语音卡进行外呼。

4、外呼通过中继，经过PSTN到达被叫端。

5、被叫设备回铃并开始振铃。

6、中继将回铃信息传送给语音卡。

7、语音卡将回铃消息传送给DeviceAdapter。

8、DeviceAdapter将消息发送给ACD。

9、ACD收到回铃消息，改变分机状态机、软电话状态机，并将消息发送给软电话客户端。

10、软电话客户端根据回铃消息改变软电话状态。

11、被叫应答，将应答消息发送给语音卡。

12、语音卡将应答消息发送给DeviceAdapter。

13、DeviceAdapter发送消息给语音卡，建立双向连接，并发送给ACD被叫应答消息。

14、ACD收到被叫应答消息，改变软电话状态机，发送消息给软电话客户端。

15、软电话客户端收到被叫应答消息，改变软电话客户端状态。

# 5、数据库结构

## 数据库设计原则

此数据库按照关系型数据库进行设计，要求对系统中的所有实体的状态变化都有记录，各个实体之间的关联关系明确，数据记录完整，不重复，不冲突，除以上要求外还遵循以下数据库设计规则：

1. 第一范式（1NF）,如果关系模式R的每个关系r的属性值都是不可分的原子值，那么称R是第一范式（First Normal Form,简记为1NF）的模式。
2. 第二范式（2NF）,如果关系模式R是1NF,且每个非主属性完全函数依赖于候选键，那么称R是第二范式（2NF）的模式。如果数据库模式中每个关系模式都是2NF,则称数据库模式为2NF的数据库模式。
3. 第三范式（3NF）,如果关系模式R是1NF，且每个非主属性都不传递依赖于R的候选键，那么称R是第三范式（3NF）的模式。如果数据库模式中每个关系模式都是3NF，则称其为3NF的数据库模式。

## 数据库表列表

数据表是记录系统中数据的实体，表的名称和功能如下

|  |  |  |
| --- | --- | --- |
| 序号 | 表名 | 说明 |
| 1 | InboundCallDetail | 呼入电话记录，记录系统中的呼入电话。 |
| 2 | OutboundCallDetail | 呼出电话记录，记录系统中呼出的电话。 |
| 3 | InboundIVRDetail | 呼入IVR记录。 |
| 4 | TransferSkillGroupDetail | 转技能组记录。 |
| 5 | TransferAgentDetail | 转坐席记录 |
| 6 | AgentStateDetail | 软电话状态明细 |
| 7 | RecordingDetail | 录音明细 |
| 8 | ConferenceDetail | 会议明细 |
| 9 | AgentLogin | 坐席登录表 |

## 数据库各表字段

### 坐席登陆表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | char | 64 | 0 | Not Null |  | uuid |
| 2 | AgentID | char | 10 | 0 | Not Null |  | 坐席ID |
| 3 | Extension | char | 10 | 0 | Not Null |  | 坐席此次登录的分机号 |
| 4 | LoginTime | DateTime | 8 | 0 | Not Null |  | 坐席登陆的时间 |
| 5 | LogoutTime | DateTime | 8 | 0 | NULL | NULL | 坐席登出的时间 |

### 坐席状态表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | char | 64 | 0 | Not Null |  | uuid |
| 2 | AgentID | char | 10 | 0 | Not Null |  | 坐席ID |
| 3 | State | int | 4 |  | NULL |  | 坐席状态类型：0-就绪，1-离席，2-事后处理 |
| 4 | Begintime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 5 | EndTime | DateTime | 8 | 0 | NULL | NULL | 结束时间 |
| 6 | AUX\_Code | int | 4 | 0 | NULL | -1 | 离席原因码 |
| 7 | AUX\_Name | char | 50 |  | NULL |  | 离席原因文字描述 |
| 8 | Remark | char | 255 |  | NULL |  | 备注 |

### 呼入电话记录表（InboundCallDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | char | 64 | 0 | Not Null |  | uuid |
| 2 | CallerNumber | Char | 20 | 0 | Null | NULL | 主叫号码 |
| 3 | CalledNumber | Char | 20 | 0 | NULL | NULL | 被叫号码 |
| 4 | StartTime | DateTime | 8 | 0 | Not Null |  | 呼入开始时间 |
| 5 | EndTime | DateTime | 8 | 0 | Not Null |  | 呼叫结束时间 |
| 6 | HangUpReason | Int | 4 | 0 | Not Null |  | 挂机原因 |
| 7 | ChannelId | Int | 4 | 0 | Null | NULL | 通道号 |
| 8 | TrunkId | Int | 4 | 0 | Null | NULL | 中继号 |
| 9 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |
| 10 | RecordingSessionID | GUID | 64 | 0 | Null | Null | 录音ID |

主键：GlobalSessionID

### 呼出电话记录表（OutboundCallDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | CallID | Char | 64 | 0 | Not Null |  | Call ID |
| 2 | ConnectID | Char | 64 | 0 | Not Null |  | 连接ID |
| 3 | Direction | Int | 1 | 0 | Not Null | 0 | 0:外呼，1：呼入 |
| 3 | Caller | Char | 20 | 0 | NULL | NULL | 主叫号码 |
| 3 | AgentID | Char | 10 | 0 | NULL | NULL | 坐席工号 |
| 4 | Called | Char | 20 | 0 | NULL | NULL | 被叫号码 |
| 5 | StartTime | DateTime | 8 | 0 | Not Null |  | 呼叫开始时间 |
| 6 | RingingTime | DateTime | 8 | 0 | NULL | NULL | 外呼振铃时间 |
| 7 | AnswerTime | DateTime | 8 | 0 | NULL | NULL | 应答时间 |
| 8 | EndTime | DateTime | 8 | 0 | Not Null |  | 呼叫结束时间 |
| 9 | HangUpReason | Int | 4 | 0 | Not Null |  | 挂机原因 |
| 10 | ChannelId | Int | 4 | 0 | Null | NULL | 通道号 |
| 11 | TrunkId | Int | 4 | 0 | Null | NULL | 中继号 |
| 13 | RecordSessionID | GUID | 74 | 0 | Null |  | 录音ID |

### 呼入IVR记录表（InboundIVRDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 3 | AnswerTime | DateTime | 8 | 0 | NULL | NULL | 应答时间 |
| 4 | EndTime | DateTime | 8 | 0 | Not Null |  | 转出时间 |
| 5 | ChannelId | Int | 4 | 0 | Null | NULL | 通道号 |
| 6 | TrunkId | Int | 4 | 0 | Null | NULL | 中继号 |
| 7 | IVRSessionID | GUID | 64 | 0 | Not Null |  | IVR唯一标识 |
| 8 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |

### 转技能组记录表（TransferSkillGroupDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | SkillGroupID | Char | 10 | 0 | Not Null |  | 技能组号 |
| 3 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 4 | AnswerTime | DateTime | 8 | 0 | NULL | NULL | 应答时间 |
| 5 | AgentID | Char | 10 | 0 | NULL | NULL | 应答坐席工号 |
| 6 | ExtensionID | Char | 10 | 0 | NULL | NULL | 应答分机号 |
| 7 | EndTime | DateTime | 8 | 0 | Not Null |  | 转出时间 |
| 8 | NoAnswerReasion | Int | 4 | 0 | Null | NULL | 无应答原因 |
| 9 | SkillSessionID | GUID | 64 | 0 | Null | NULL | 转技能组唯一标识 |
| 10 | IVRSessionID | GUID | 64 | 0 | Not Null |  | IVR唯一标识 |
| 11 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |

### 转坐席记录表（TransferAgentDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 3 | RingTime | DateTime | 8 | 0 | NULL | NULL | 振铃时间 |
| 4 | AgentID | Char | 10 | 0 | NULL | NULL | 坐席工号 |
| 5 | ExtensionID | Char | 10 | 0 | NULL | NULL | 分机号 |
| 6 | AnswerTime | DateTime | 8 | 0 | NULL | NULL | 应答时间 |
| 7 | EndTime | DateTime | 8 | 0 | Not Null |  | 挂机时间 |
| 8 | NoAnswerReasion | Int | 4 | 0 | Null | NULL | 无应答原因 |
| 9 | SkillSessionID | GUID | 64 | 0 | Null | NULL | 转技能组唯一标识 |
| 10 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |

### 软电话状态明细记录表（AgentStateDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 3 | AgentID | Char | 10 | 0 | Not Null |  | 坐席工号 |
| 4 | ExtensionID | Char | 10 | 0 | NULL | NULL | 分机号 |
| 5 | EndTime | DateTime | 8 | 0 | Not Null |  | 结束时间 |
| 6 | AgentSessionID | GUID | 64 | 0 | Not Null |  | 全局标识 |
| 7 | RecordingSessionID | GUID | 64 | 0 | Null |  | 录音ID |

### 录音明细记录表（RecordingDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | CallerNumber | Char | 20 | 0 | Null | Null | 主叫号码 |
| 3 | CalledNumber | Char | 20 | 0 | Null | Null | 被叫号码 |
| 4 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 5 | AgentID | Char | 10 | 0 | NULL | NULL | 坐席工号 |
| 6 | ExtensionID | Char | 10 | 0 | NULL | NULL | 分机号 |
| 7 | EndTime | DateTime | 8 | 0 | Not Null |  | 挂机时间 |
| 8 | ChannelID | Int | 4 | 0 | Null | Null | 录音通道号 |
| 9 | TrunkID | Int | 4 | 0 | Null | Null | 录音设备号 |
| 10 | RecordSessionID | GUID | 64 | 0 | Not Null |  | 录音全局标识 |
| 11 | GlobalSessionID | GUID | 64 | 0 | Not Null |  | 全局通话标识 |

### 会议明细记录表（ConferenceDetail）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 大小 | 精度 | 可为空 | 默认值 | 说明 |
| 1 | Id | bigInt | 8 | 0 | Not Null |  | 如果数据量过大，可能会超过此数据类型的最大值。 |
| 2 | ConferenceID | Char | 10 | 0 | Not Null | Null | 会议号码 |
| 3 | StartTime | DateTime | 8 | 0 | Not Null |  | 开始时间 |
| 4 | ConferenceSessionID | GUID | 64 | 0 | Not Null |  | 会议唯一标识 |
| 5 | EndTime | DateTime | 8 | 0 | Not Null |  | 挂机时间 |
| 6 | RecordingSessionID | GUID | 64 | 0 | NuLL |  | 录音ID |

## 数据库结构图

数据库中的各表主要关系如下：

